

IN THE  
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Sally Elaine Saffer

Confirmation No.: 6758

Application No.: 10/068,466

Examiner: Belix M. Ortiz

Filing Date: 02/05/2002

Group Art Unit: 2164

Title: Operational Data Store

Mail Stop Appeal Brief-Patents  
Commissioner For Patents  
PO Box 1450  
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on June 16, 2005.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( ) one month	\$120.00
( ) two months	\$450.00
( ) three months	\$1020.00
( ) four months	\$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of \$500.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: 8/16/2005

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number \_\_\_\_\_ on \_\_\_\_\_

Number of pages:

Typed Name: Amy T. Comeau

Signature: Amy T. Comeau

Respectfully submitted,

Sally Elaine Saffer

By Robert Plotkin

Robert Plotkin, Esq.

Attorney/Agent for Applicant(s)  
Reg. No. 43,861

Date: August 16, 2005

Telephone No.: (978) 318-9914

ATTORNEY'S DOCKET NO: 200302250-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Sally Elaine Saffer  
Serial No: 10/068,466  
Filed: February 5, 2002  
For: Operational Data Store

Examiner: Belix M. Ortiz  
Art Unit: 2164

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to the Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450 on the 16th day of August, 2005.



Amy T. Comeau

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

APPELLANTS' BRIEF ON APPEAL

This is an appeal pursuant to 35 U.S.C. § 134 from the Examiner's decision rejecting claims 1-4, 8-22, 29-35, 39-53, 60-63, 65-67, and 69-73 as set forth in the Final Office Action of March 7, 2005 and the Advisory Action of June 2, 2005.

**REAL PARTY IN INTEREST**

The real party in interest is Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

**RELATED APPEALS AND INTERFERENCES**

Applicant's attorney knows of no related pending appeals or interferences.

**STATUS OF CLAIMS**

Claims 1-73 are pending in this application.

Claims 1-4, 8-22, 29-35, 39-53, 60-63, 65-67, and 69-73 stand rejected and are the subject of this appeal. More specifically, claims 1-4, 8-13, 29-35, 39-44, 60-63, 65-67, and 69 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Kawai (U.S. Pat. No. 5,717,924). Claims 70-73 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Battas et al. (U.S. Pat. No. 6,757,706). Claims 14-22 and 45-53 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kawai in view of Kessler et al. (U.S. Pat. No. 5,761,706).

**STATUS OF AMENDMENTS**

No after-final amendments have been filed in this case.

**SUMMARY OF CLAIMED SUBJECT MATTER**

Independent claim 1 is directed to an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5) comprising an insert table (FIG. 1, element 32; p. 6, lines 1-5, 22-25) for storing new data; a history table (FIG. 1, element 30; p. 6, lines 1-5, 22-25) for storing historical data; and transfer logic (FIGS. 3A and 5, element 56; p. 6, lines 22-25; p. 7, lines 27-29, p. 8, lines 23-25; p. 14, lines 19-21) for periodically transferring new data from the insert table to the history table.

Independent claim 32 is directed to a method for maintaining an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5), the method comprising inserting new data into an insert table (FIG. 1, element 32; p. 6, lines 1-5, 22-25; FIG. 5, element 122; p. 8, lines 23-25); and periodically transferring data from the insert table to a history table (FIGS. 3A and 5, element 56; p. 6, lines 22-25; p. 7, lines 27-29, p. 8, lines 23-25; p. 14, lines 19-21).

Independent claim 60 is directed to a method for operating an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5), the method comprising creating a new partition in a composite-partitioned history table (FIG. 7, step 206; p. 15, lines 17-19); creating a partitioned temporary table (FIG. 7, step 210; p. 15, lines 22-28); filling the temporary table with data from an

insert table (FIG. 7, step 210; p. 15, lines 22-28); exchanging the temporary table with the new partition (FIG. 7, step 214; p. 16, lines 4-7); and receiving a query and applying the query to both the history table and the insert table (p. 4, lines 8-9; p. 6, line 26 - p. 7, line 10; FIG. 2, elements 58, 60).

Independent claim 62 is directed to an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5) comprising means for inserting new data into an insert table (FIG. 1, element 32; p. 6, lines 1-5, 22-25; FIG. 5, element 122; p. 8, lines 23-25); means for periodically transferring data from the insert table to a history table (FIGS. 3A and 5, element 56; p. 6, lines 22-25; p. 7, lines 27-29, p. 8, lines 23-25; p. 14, lines 19-21); and means for applying a database query to both the history table and the insert table (p. 4, lines 8-9; p. 6, line 26 - p. 7, line 10; FIG. 2, elements 58, 60).

Independent claim 65 is directed to an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5) comprising means for creating a new partition in a composite-partitioned history table (FIG. 7, step 206; p. 15, lines 17-19); means for creating a partitioned temporary table (FIG. 7, step 210; p. 15, lines 22-28); means for filling the temporary table with data from an insert table (FIG. 7, step 210; p. 15, lines 22-28); means for exchanging the temporary table with the new partition (FIG. 7, step 214; p. 16,

lines 4-7); and means for receiving a database query and applying said query to both the history table and the insert table (FIG. 7, step 214; p. 16, lines 4-7).

Independent claim 66 is directed to a computer program product for operating an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5), the computer program product comprising a computer usable medium having computing reading code thereon (p. 18, lines 11-19), including program code which: inserts new data into an insert table (FIG. 1, element 32; p. 6, lines 1-5, 22-25; FIG. 5, element 122; p. 8, lines 23-25); periodically transfers data from the insert table to a history table (FIGS. 3A and 5, element 56; p. 6, lines 22-25; p. 7, lines 27-29, p. 8, lines 23-25; p. 14, lines 19-21); and applies a database query to both the history table and the insert table (p. 4, lines 8-9; p. 6, line 26 - p. 7, line 10; FIG. 2, elements 58, 60).

Independent claim 69 is directed to a computer program product for operating an operational data store (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5), the computer program product comprising a computer usable medium having computer readable code thereon (p. 18, lines 11-19), including program code which: creates a new partition in a composite-partitioned history table (FIG. 7, step 206; p. 15, lines 17-19); creates a partitioned temporary table (FIG. 7, step 210; p. 15, lines 22-28); fills the temporary table with data from

an insert table (FIG. 7, step 210; p. 15, lines 22-28); exchanges the temporary table with the new partition (FIG. 7, step 214; p. 16, lines 4-7); and receives queries and applies said queries to both the history table and the insert table (FIG. 7, step 214; p. 16, lines 4-7).

Independent claim 70 is directed to a system for producing a desired level of service in a mixed workload environment, the system comprising: a high-speed insert operational data store (ODS) (FIG. 1, element 24; p. 5, line 18 - p. 6, line 5); a throttler for throttling selected transactions to the ODS (FIG. 2, element 44; p. 9, line 4 - p. 11, line 3); and an aggregator for accumulating transactions into batches and inserting each of the batches into the ODS using a single database transaction per batch (FIG. 2, lines 48-50; p. 11, line 25 - p. 13, line 24).

Independent claim 72 is directed to a method for producing a desired level of service in a mixed workload environment, the method comprising: insert transactions into an operational data store (ODS) at a high-speed (FIG. 1, element 32; p. 6, lines 1-5, 22-25; FIG. 5, element 122; p. 8, lines 23-25); throttling selected transactions to the ODS (FIG. 2, element 44; p. 9, line 4 - p. 11, line 3); accumulating transactions into batches (FIG. 2, lines 48-50; p. 11, line 25 - p. 13, line 24); and inserting each of the batches into

the ODS using a single database transaction per batch (FIG. 2, lines 48-50; p. 11, line 25 - p. 13, line 24).

**GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

The grounds of rejection for review are:

- (1) the rejection of claims 1-4, 8-13, 29-35, 39-44, 60-63, 65-67, and 69 under 35 U.S.C. § 102(e) as being anticipated by Kawai (U.S. Pat. No. 5,717,924);
- (2) the rejection of claims 70-73 under 35 U.S.C. § 102(e) as being anticipated by Battas et al. (U.S. Pat. No. 6,757,706); and
- (3) the rejection of claims 14-22 and 45-53 under 35 U.S.C. § 103(a) as being unpatentable over Kawai (U.S. Pat. No. 5,717,924) in view of Kessler et al. (U.S. Pat. No. 5,761,706).

**ARGUMENT**

**Rejection of Claims 1-4, 8-13, 29-35, 39-44, 60-63, 65-67, and 69  
under 35 U.S.C. § 102(e) (Kawai)**

Claims 1-4, 8-13, 29-35, 39-44, 60-63, 65-67, and 69 stand rejected under 35 U.S.C. 102(e) as being anticipated by Kawai (U.S. Pat. No. 5,717,924). The Examiner has provided different grounds of



rejection for different subsets of these claims. Each distinct ground of rejection will be argued separately, as indicated by the subheadings below.

**Claims 1-4, 8-13, 29-35, 39-44, 62-63, and 66-67**

The Final Office Action states that Kawai teaches an operational data store comprising an insert table for storing new data (referring to FIG. 10A, step 320); a history table for storing historical data (referring to FIG. 10A, step 328); and transfer logic for periodically transferring new data from the insert table to the history table (referring to FIG. 10A).

FIG. 10A does not, however, teach these elements of claim 1 of the present application. Rather, FIG. 10A of Kawai teaches a method for copying *old table data* into a new table. More specifically, the method shown in FIG. 10A of Kawai includes three steps: (1) creating a *new database table* with a new name; (2) copying data *from an old table to the new table*; and (3) dropping the old table from the database.

Claim 1 of the present application recites precisely the opposite: "transfer logic for periodically transferring *new data* from the insert table to the history table," which stores historical (old) data. In other words, Kawai teaches copying *old table data* into a new table, while claim 1 of the present application recites

copying new table data into an old table. Therefore, claim 1 of the present application patentably distinguishes over Kawai.

It is well-settled that anticipation of a claim under 35 U.S.C. § 102 only exists where a single prior art reference identically sets forth each and every claim limitation.<sup>1</sup> Both the Final and Advisory Office Actions ignore express limitations of claim 1 requiring that the insert table store "new data" and that the history table store "historical data." Rather, the Final and Advisory Actions gloss over these limitations by relying on the assertion that "Kawai teaches two [different] table[s], one with new data and another table with old data, and *copy data from one table to another table*" (emphasis added). Even if it is true that Kawai teaches copying data from one table to another table, claim 1 of the present application does not generally recite copying data from any table to any other table. Rather, it specifically recites copying

---

<sup>1</sup> See, e.g., *Mehl/Biophile International Corp. v. Milgraum*, 192 F.3d 1362, 1365, 52 USPQ2d 1303, 1306 (Fed. Cir. 1999) ("To anticipate, a single reference must teach every limitation of the claimed invention."); *General Electric Co. v. Nintendo Co., Ltd.*, 179 F.3d 1350, 1356, 50 USPQ2d 1910, 1915 (Fed. Cir. 1999) ("A judgement of invalidity for anticipation requires that a single prior art reference disclose every limitation in a patent claim."); *Rockwell International Corp. v. United States*, 147 F.3d 1358, 1363, 47 USPQ2d 1027, 1031 (Fed. Cir. 1998) ("Anticipation under 35 U.S.C. Section 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention.").

new data from an insert table into a historical table that stores *historical* (old) data. Because Kawai discloses precisely the opposite - copying *old* table data into a *new* table - Kawai fails to disclose express limitations of claim 1 of the present application and therefore fails to anticipate claim 1.

Claim 1 of the present application, therefore, patentably distinguishes over Kawai. Claims 2-4, 8-13, and 29-31 depend, either directly or indirectly, from claim 1 and therefore patentably distinguish over claim 1 for at least the same reason. Claims 32, 62, 66 include substantially the same relevant limitations as claim 1 and therefore patentably distinguish over Kawai for at least the same reasons. Claims 33-35, 39-44, 63, and 67 depend, either directly or indirectly, from claims 32, 62, and 66, and therefore patentably distinguish over Kawai for at least the same reason.

### **Claims 60-61, 65, and 69**

With respect to claim 60, the Final Office Action states that Kawai teaches a method for operating an operational data store comprising creating a new partition in a composite-partitioned history table (referring to FIG. 7); creating a partitioned temporary table (referring to FIG. 10B, step 332); filling the temporary table with data from an insert table (referring to FIG. 10B, step 336); exchanging the temporary table with the new partition (referring to FIG. 10B, step 348); and receiving a query

and applying the query to both the history table and the insert table (referring to col. 2, lines 19-21 and col. 4, lines 29-42).

Kawai does not, however, teach "creating a new partition in a composite-partitioned history table," as expressly required by claim 60 of the present application. Support for this element is provided, for example, at page 15, lines 18-19 of the present application, which states that in step 206 of FIG. 6, "a new partition is created in the historical table, partitioned by range and sub-partitioned by the number of database server instances." A partition that is sub-partitioned in this manner is an example of a composite-partitioned table.

Both the Final and Advisory Office Actions state that "Kawai teaches table with partition and partitioned table with more partition (see Kawai, figures 7 and 8)." This is simply not correct. Partitioning involves decomposing a database table into smaller pieces called "partitions."<sup>2</sup> Composite partitioning involves subdividing partitions into subpartitions. FIGS. 7 and 8 of Kawai do not show composite partitioning; they merely show related tables in a relational database. For example, FIG. 7 of Kawai shows a "student" table 172 and a related "major" table 176. Even if it is

---

<sup>2</sup> See, e.g., Microsoft Computer Dictionary 392 (5th ed. 2002) (defining "partition" as "a subset of a database table or file").

assumed for purposes of argument that the student table 172 and major table 176 are two *single* partitions, neither FIG. 7 nor any other portion of Kawai discloses *composite* partitioning, i.e., the division of partitions into subpartitions. Kawai, therefore, fails to teach an express limitation of claim 60, which therefore patentably distinguishes over Kawai. Claim 61 depends from claim 60 and therefore patentably distinguishes over Kawai for at least the same reason. Claims 65 and 69 include the same relevant limitations as claim 60 and therefore patentably distinguish over Kawai for at least the same reason.

Rejection of Claims 70-73 under 35 U.S.C. § 102(e) (Battas)

Claims 70-73 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Battas et al. (U.S. Pat. No. 6,757,706). With respect to claim 70, the Final Office Action states that Battas teaches a system for producing a desired level of service in a mixed workload environment (referring to col. 20, lines 31-34), comprising a high-speed insert operational data store (ODS) (referring to col. 15, lines 34-37); a throttler for throttling selected transactions to the ODS (referring to col. 15, lines 41-45); and an aggregator for accumulating transactions into batches and inserting each of the batches into the ODS using a single database transaction per batch (referring to col. 15, lines 41-46).

Battas does not, however, teach "a throttler for throttling selected transactions to the ODS," as required by claim 70 of the present application. Support for this limitation may be found in the present application, for example, at p. 4, lines 16-20; p. 6, lines 16-17; p. 7, lines 24-25; p. 8, lines 1-6; p. 9, lines 4-28; and p. 10, lines 1-24. For example, the present application provides one example of throttling selected transactions at p. 98, lines 7-10, stating that "[t]o achieve a desired level of service, different 'classes' of transactions may be independently throttled. That is, to provide the desired performance of the high-speed inserts, queries may need to be slowed, so that the higher-priority insert transactions can execute at full speed" (emphasis added).

To support the rejection of claims 70-73, the Final Office Action states that Battas teaches:

4) parallel message router and inserter service 124 for high performance, high-volume routing, and insertion of data into the ODS and other ZLE services and applications; 5) event capture service 125 for supporting the enterprise-wide business; and 6) extraction, transformation, and load (ETL) service 126, to move bulk data into and out of the ODS and across database and platform boundaries. The bulk data extraction service is for uploading aggregated batched transactions back into the ODS, and for moving huge volumes of data quickly out of the ODS to production systems, business intelligence systems.

Nothing in the above-quoted passage, however, indicates how Battas teaches "throttling selected transactions to the ODS" as

required by claim 70. As described in the present application, for example, at page 9, lines 8-10, one example of throttling selected transactions is slowing the processing of query transactions so that higher-priority insert transactions can execute at full speed. Neither the Final Office Action nor the Advisory Office Action points to anything in Battas which teaches or suggests slowing or otherwise throttling the processing of selected transactions, and Battas does not in fact provide any such teaching or suggestion.

Claim 70 of the present application therefore patentably distinguishes over Battas. Claim 72 includes the same relevant limitations as claim 70 and therefore patentably distinguishes over Battas for at least the same reasons. Claims 71 and 73 depend from claims 70 and 72, respectively, and therefore patentably distinguish over Battas for at least the same reason.

Rejection of Claims 14-22 and 45-53 under 35 U.S.C. § 103(a)

(Kawai in view of Kessler)

Claims 14-22 and 45-53 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kawai in view of Kessler et al. (U.S. Pat. No. 5,761,706). Claims 14-22 and 45-53 are dependent claims which depend from independent claims discussed above and therefore include at least the same limitations. For the reasons described above, Kawai fails to teach at least one express limitation of each of claims 14-22 and 45-53. The Final Office Action fails to point

out how Kessler teaches or suggests any of these limitations, and Kessler does not in fact teach or suggest any of these limitations. Therefore, claims 14-22 and 45-53 patentably distinguish over the combination of Kawai and Kessler for at least the same reasons provided above.

### CONCLUSIONS

The Examiner's rejections of claims 1-4, 8-22, 29-35, 39-53, 60-63, 65-67, and 69-73 should be reversed for the reasons stated above.

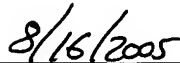
If this Brief is not considered timely filed and if a request for extension of time is otherwise absent, applicant hereby requests any extension of time. Please charge any fees or make any credits, to Deposit Account No. 08-2025.

Respectfully submitted,



Robert Plotkin, Esq.

Reg. No. 43,861



Date

Robert Plotkin, P.C.  
91 Main Street, Suite 204  
Concord, MA 01742-2527  
Tel: (978) 318-9914  
Fax: (978) 318-9060



**APPENDIX A: CLAIMS ON APPEAL**

Claim 1. An operational data store, comprising:

an insert table for storing new data;

a history table for storing historical data; and

transfer logic for periodically transferring new data from the insert table to the history table.

Claim 2. The operational data store of Claim 1, wherein the history table is partitioned.

Claim 3. The operational data store of Claim 2, wherein the history table is partitioned by range.

Claim 4. The operational data store of Claim 2, wherein each partition is further sub-partitioned.

Claim 8. The operational data store of Claim 1, the transfer logic comprising:

a secondary table;

fill logic for filling the secondary table with selected data from the insert table; and

secondary transfer logic for transferring the secondary table into the history table, the selected data thereby being transferred into the history table.

Claim 9. The operational data store of Claim 8, wherein the history table has an indexing scheme, the secondary transfer logic further comprising:

indexing logic for applying the history table indexing scheme to the secondary table.

Claim 10. The operational data store of Claim 9, wherein the indexing logic applies the history table indexing scheme to the secondary table prior to transferring the secondary table into the history table.

Claim 11. The operational data store of Claim 8, the secondary transfer logic further comprising:

table logic for creating a new partition the history table, the new partition for swapping with the secondary table.

Claim 12. The operational data store of Claim 11, wherein the secondary transfer logic swaps the secondary table and the new partition by exchanging respective pointers.

Claim 13. The operational data store of Claim 1, further comprising:

a query engine for applying a database query to both the history table and the insert table.

Claim 14. The operational data store of Claim 1, further comprising:

an aggregation buffer for accumulating new data; and

an aggregator for batching the accumulated data and transferring the batched data into the insert table with a single database access.

Claim 15. The operational data store of Claim 14, wherein the aggregator transfers a batch of new data into the insert table when the batch surpasses a maximum size.

Claim 16. The operational data store of Claim 15, wherein batch size is measured according to a number data bytes.

Claim 17. The operational data store of Claim 15, wherein batch size is measured according to a number of records.

Claim 18. The operational data store of Claim 15, wherein the maximum size is configurable.

Claim 19. The operational data store of Claim 14, wherein the aggregator transfers batches of new data into the insert table at regular intervals, defined by a given period.

Claim 20. The operational data store of Claim 19, wherein the period is configurable.

Claim 21. The operational data store of Claim 14, wherein the aggregator transfers batches of new data into the insert table when the aggregation buffer surpasses a given maximum buffer size.

Claim 22. The operational data store of Claim 21, wherein the maximum buffer size is configurable.

Claim 29. The operational data store of Claim 1, wherein data from the insert table is transferred to the history table at regular intervals.

Claim 30. The operational data store of Claim 29, wherein the intervals are configurable.

Claim 31. The operational data store of Claim 29, wherein the intervals are different for different tables.

Claim 32. A method for maintaining an operational data store, comprising:

Inserting new data into an insert table;

Periodically transferring data from the insert table to a history table.

Claim 33. The method of Claim 32 further comprising:

Partitioning the history table.

Claim 34. The method of Claim 33, wherein the history table is partitioned according to range.

Claim 35. The method of Claim 33, further comprising:

sub-partitioning each partition.

Claim 39. The method of Claim 32, further comprising:

creating a secondary table;

filling the secondary table with selected data from the insert table; and

transferring the secondary table into the history table, the selected data thereby being transferred into the history table.

Claim 40. The method of Claim 39, wherein the history table has an indexing scheme, the method further comprising:

applying the history table indexing scheme to the secondary table.

Claim 41. The method of Claim 40, wherein the history table indexing scheme is applied to the secondary table prior to transferring the secondary table into the history table.

Claim 42. The method of Claim 39, further comprising:

creating a new partition in the history table, wherein the secondary table is transferred by being swapped with the new partition.

Claim 43. The method of Claim 42, wherein the secondary table and new partition are swapped by exchanging respective pointers.

Claim 44. The method of Claim 32, further comprising:

applying a database query to both the history table and the insert table.

Claim 45. The method of Claim 32, further comprising:

aggregating new data into batches; and

inserting the batched new data into the insert table with a single database access.

Claim 46. The method of Claim 45, wherein a batch of new data is transferred into the insert table when the batch surpasses a maximum size.

Claim 47. The method of Claim 46, wherein batch size is measured according to a number data bytes.

Claim 48. The method of Claim 46, wherein batch size is measured according to a number of records.

Claim 49. The method of Claim 46, wherein the maximum size is configurable.

Claim 50. The method of Claim 45, wherein batches of new data are transferred into the insert table at regular intervals, defined by a given period.

Claim 51. The method of Claim 50, wherein the period is configurable.

Claim 52. The method of Claim 45, further comprising:

aggregating the batches of new data in an aggregation buffer, wherein the batches are transferred into the insert table when the aggregation buffer surpasses a given maximum buffer size.



Claim 53. The method of Claim 52, wherein the maximum buffer size is configurable.

Claim 60. A method for operating an operational data store, comprising:

- creating a new partition in a composite-partitioned history table;

- creating a partitioned temporary table;

- filling the temporary table with data from an insert table;

- exchanging the temporary table with the new partition; and

- receiving a query and applying the query to both the history table and the insert table.

Claim 61. The method of claim 60, further comprising:

- creating a new partition in the insert table based on values from an existing partition; and

- dropping the existing partition.

Claim 62. An operational data store, comprising:

means for inserting new data into an insert table;

means for periodically transferring data from the insert table to a history table; and

means for applying a database query to both the history table and the insert table.

Claim 63. The operational data store of Claim 62, further comprising:

means for batching new data; and

means for inserting the batched new data into the insert table with a single database access.

Claim 65. An operational data store, comprising:

means for creating a new partition in a composite-partitioned history table;

means for creating a partitioned temporary table;

means for filling the temporary table with data from an insert table;

means for exchanging the temporary table with the new partition; and

means for receiving a database query and applying said query to both the history table and the insert table.

Claim 66. A computer program product for operating an operational data store, the computer program product comprising a computer usable medium having computing reading code thereon, including program code which:

inserts new data into an insert table;

periodically transfers data from the insert table to a history table; and

applies a database query to both the history table and the insert table.

Claim 67. The computer program product of Claim 66, wherein the program code further:

batches new data; and

inserts the batched new data into the insert table with a single database access.

Claim 69. A computer program product for operating an operational data store, the computer program product comprising a

computer usable medium having computer readable code thereon,  
including program code which:

creates a new partition in a composite-partitioned history  
table;

creates a partitioned temporary table;

fills the temporary table with data from an insert table;

exchanges the temporary table with the new partition; and

receives queries and applies said queries to both the  
history table and the insert table.

Claim 70. A system for producing a desired level of service in  
a mixed workload environment, comprising:

a high-speed insert operational data store (ODS);

a throttler for throttling selected transactions to the  
ODS; and

an aggregator for accumulating transactions into batches  
and inserting each of the batches into the ODS using a single  
database transaction per batch.

Claim 71. The system of Claim 70, wherein the mixed workload  
environment includes at least two of archiving, OLTP queries, DSS

queries, high-speed inserts, backup processes and extract/translate/load transactions.

Claim 72. A method for producing a desired level of service in a mixed workload environment, comprising:

insert transactions into an operational data store (ODS)  
at a high-speed;

throttling selected transactions to the ODS;

accumulating transactions into batches; and

inserting each of the batches into the ODS using a single database transaction per batch.

Claim 73. The method of Claim 72, wherein the mixed workload environment includes at least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup processes and extract/translate/load transactions.

**APPENDIX B: EVIDENCE**

No evidence is submitted in support of this Appeal Brief.

**APPENDIX C: RELATED PROCEEDINGS**

Applicant's attorney knows of no related pending appeals or interferences.